REMARKS

Claims 1-7, 10 and 12-17 remain pending in the application. Applicant respectfully requests reconsideration in light of the amendments to the claims and the remarks provided herein.

Applicant's invention is directed toward a cache controller for a microprocessor that executes multiple tasks concurrently. The cache controller has a cache memory partitioned into multiple regions that correspond to the multiple tasks. The cache controller receives addresses from the microprocessor, searches all of the data in all of the regions and judges whether data corresponding to the received addresses are stored in the cache memory.

If data corresponding to a received address is stored in the cache memory of the cache controller, a "cache hit" judgment is made. If data corresponding to a received address is not stored in the cache memory of the cache controller, a "miss" judgment is made. When a miss judgment is made, data may be moved from main memory to the region of cache memory associated with the particular task that requested the data.

Applicant's cache controller allows each task to have full access to all of the data stored in cache memory. The cache controller also moves data from main memory to the cache region associated with the task reducing the probability of any one single task negatively impacting the hit rate of any other task being processed concurrently.

Claims 1-3, 10, and 12-17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Handy* ("The Cache Memory Book") in view of *Suh et al* ("Dynamic Cache Partitioning for Simultaneous Multithreading Systems") further in view of *Stevens* (U.S. Pat. No. 6,745,292).

Handy discloses a cache controller that receives an address from a microprocessor

(*Handy*, Figure 1.6). The cache controller manages a cache directory and a cache memory (*Handy*, Page 12, Line 15 – Page 13, Line 1). The cache directory and memory store data and addresses based on the cache strategy or policy (*Handy*, Page 13, Lines 5-13). The cache controller receives addresses from the microprocessor and determines whether the data is in the cache memory. If the data is not available in the cache memory, the cache controller may acquire the data from the main memory and store it in the cache memory.

Handy, as the Office Action acknowledges, does not disclose a judging unit for judging whether the data stored at the received address is in the cache memory, by searching all data in a plurality of regions (Office Action, Page 3, Lines 9-10).

Suh discloses a method for dynamically partitioning the cache memory of multithreading systems (*Suh*, Abstract). *Suh's* method compares the miss statistics of each thread and computes a marginal gain of allocating another data block of memory for each of the threads (*Suh*, Section 3.1, Paragraph 1, Section 3.2 Paragraphs 1-2). The marginal gain statistics are used to partition the cache memory (*Suh* Section 3.3 and 3.4).

As acknowledged by the Office Action, *Suh* like *Handy* fails to disclose a judging unit for judging whether the data stored at the received address is in the cache memory, by searching all data in a plurality of regions (Office Action, Page 3, Lines 9-10)

Steven's disclosure is directed toward a computer system with a cache memory shared by multiple processors, (*Stevens*, Abstract). The cache memory is divided into n-data sets each partitioned into multiple regions, (*Stevens*, Figure 3, Column 3, Lines 53-56). The n data sets are logical groupings of data associated with corresponding tasks. Cache searches are mapped by task and searches may occur within the data set over multiple regions, (*Stevens*, Figure 3). Cache searches over one complete data set, thus, occur over each one of *Steven's* cache regions,

(*Stevens*, Figure 3).

Steven's searches are over a particular data set, a subset of all the data in the *Steven's* regions. *Stevens* is notably silent regarding searching all of the data in all of the regions.

Claims 1-3, 10 and 12-17 now recite "a judging unit (step) operable to judge whether the data stored at the received address is stored in the cache memory, by searching all data in all of the plurality of regions in the cache memory". The claim language of claims 1-3, 10 and 12-17 has been amended to further distinguish over the prior art of record. Support for the amendment may be found in the specification, (Application, Page 26, Section 6).

The Office Action admits that *Handy* and *Suh* fail to disclose the judging unit recited in the claims, (Office Action, Page 3, Lines 9-10). The Office Action, however, asserts that *Stevens* discloses the judging unit because *Stevens* teaches "judging whether the requested data is stored in the cache memory by searching all of the plurality of regions in the cache memory", (Office Action, Page 3, Lines 9-13). . Applicant respectfully traverses.

The Office Action relies on two passages (i.e. the abstract and column 1 lines 36-40) in its assertion that *Stevens* teaches Applicant's judging unit, (Office Action Page 3, Lines 13-15). Yet neither of these passages mentions a judging unit. The first passage (abstract) suggests that *Stevens* device has access to all regions on hits and the second passage (background) provides a definition of "hit". Neither passage discloses or suggests searching all of the data in all of the regions.

In fact, *Steven's* (hit or miss) judgment is not made by searching all of the data in all of the plurality of regions. *Steven's* judgment is made by searching one of n sets of data having data in a plurality of regions. Specifically, *Stevens* teaches that one of the n sets (i.e. set j) of data is comprised of 16 partitions divided into four regions, (*Stevens*, Column 3, Lines 59-61).

Each of the four regions is controlled by a corresponding processor, (*Stevens*, Column 3, Lines 61-63). Each of the processors has access to all four regions of set j, but each of the processors can only allocate to the corresponding region of set j, (*Stevens*, Column 3, Lines 64-66).

Steven's thus teaches making (hit or miss) judgments by searching a plurality of regions of one set (set j) of data and allocating data to regions that corresponds with each microprocessor. *Steven's,* like *Handy* and *Suh,* fails to disclose or suggest searching all of the data in all of the plurality of regions making claims 1-3, 10 and 12-17 patentable over any combination of *Handy, Suh* and *Stevens*.

Searching all of the data in all of the regions of cache memory is an important feature of Applicant's cache controller. This feature makes cache controller search procedures largely independent of the particular task requesting the search. Task data for a particular task does not have to be grouped into cache sets as in *Stevens*. Particular tasks do not have to track the cache use of other tasks as in *Stevens*. This feature simplifies the data caching process and the data caching controller.

Notwithstanding the failure of the *Handy, Suh,* and *Stevens* combination to teach each and every element of the claimed invention, the combination is unobvious because it renders the *Suh* device inoperable for its intended purpose.

> [I]t is generally settled that the change in prior art device which makes the device inoperable for its intended purpose cannot be considered to be an obvious change.
>
> *Hughes Aircraft Co. v. United States*, 215 U.S.P.Q. 787, (Ct.Cl. Trial Div. 1982)

The purpose of *Suh's* device is to create an optimal partition of cache memory by iteratively increasing the partition for the thread that will benefit the most, (*Suh*, page 432). *Suh*

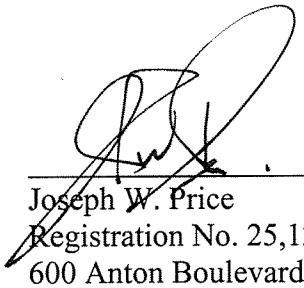
determines the optimum partition by calculating total misses over time (*Suh*, page 432, equation 1). *Steven's,* in contrast, partitions the cache memory by the individual processor that fetched the data, (*Stevens*, Column 1, Lines 61-63). Thus incorporating *Stevens* judging function with *Suh* as suggested by the Office Action, (Office Action, Page 3, Lines 15-20) would result in a device having non-optimal partitioning, the stated purpose of the *Suh* device, rendering the *Suh* device inoperable for its intended purpose and making the combination unobvious.

For the reasons stated above Applicant believes the application is now in condition for allowance and early notification of the same is respectfully requested.

If the Examiner believes that a telephone interview will help further the prosecution of the case, the undersigned attorney can be contacted at the listed telephone number.

Very truly yours,

SNELL & WILMER L.L.P.



Joseph W. Price
Registration No. 25,124
600 Anton Boulevard, Suite 1400
Costa Mesa, California 92626-7689
Telephone: (714) 427-7420
Facsimile: (714) 427-7799